Minimum Margin Loss for Deep Face Recognition

Xin Wei[a,*], Hui Wang[a], Bryan Scotney[b], Huan Wan[a]

[a]*School of Computing, Ulster University at Jordanstown, BT370QB, UK*
[b]*School of Computing, Ulster University at Coleraine, BT521SA UK*

Abstract

Face recognition has achieved great success owing to the fast development of deep neural networks in the past few years. Different loss functions can be used in a deep neural network resulting in different performance. Most recently some loss functions have been proposed, which have advanced the state of the art. However, they cannot solve the problem of *margin bias* which is present in class imbalanced datasets, having the so-called long-tailed distributions. In this paper, we propose to solve the margin bias problem by setting a minimum margin for all pairs of classes. We present a new loss function, Minimum Margin Loss (MML), which is aimed at enlarging the margin of those overclose class centre pairs so as to enhance the discriminative ability of the deep features. MML, together with Softmax Loss and Centre Loss, supervises the training process to balance the margins of all classes irrespective of their class distributions. We implemented MML in Inception-ResNet-v1 and conducted extensive experiments on seven face recognition benchmark datasets, MegaFace, FaceScrub, LFW, SLLFW, YTF, IJB-B and IJB-C. Experimental results show that the proposed MML loss function has led to new state of the art in face recognition, reducing the negative effect of margin bias.

Keywords: Deep learning, Convolutional neural networks, Face recognition, Minimum Margin Loss

*Corresponding author
Email address: wei-x@ulster.ac.uk (Xin Wei)

Table 1: Statistics for recent public available large-scale face datasets.

	MS-Celeb-1M	VGGFace2	MegaFace	CASIA
#Identities	100K	9K	672K	11K
#Images	10M	3M	5M	0.5M
Avg per Person	105	323	7	47

1. Introduction

In the past ten years, deep neural network (DNN) based methods have achieved great progress in various computer vision tasks, including face recognition [1], person re-identification [2], object detection [3] and action recognition [4]. The progress on face recognition is particularly remarkable due largely to two important factors – larger face datasets and better loss functions.

The quantity and quality of the face datasets used for training directly influence the performance of a DNN model in face recognition. Currently, there are a few large-scale face datasets that are publicly available, for example, MS-Celeb-1M [5], VGGFace2 [6], MegaFace [7] and CASIA WebFace [8]. As shown in Table 1, CASIA WebFace consists of 0.5M face images; VGGFace2 contains totally 3M face images but only from 9K identities; MS-Celeb-1M and MegaFace both contain more images and more identities, thus should have greater potential for training a better DNN model. However, both MS-Celeb-1M and MegaFace have the problem of long-tailed distribution [9], which means a minority of people owns a majority of face images and a large number of people have very limited face images. Using datasets with long-tailed distribution, the trained model tends to overfit the classes with rich samples thus weakening the generalisation ability on the long-tailed portion [9]. Specifically, the classes with rich samples tend to have a relatively large margin between their class centres; conversely, the classes with limited samples tend to have a relatively small margin between their class centres as they only occupy a small region in space and are thus easy to be compressed. This *margin bias* problem is due to long-tailed class distribution, which leads to performance drop on face recognition [9].

Besides the training set and its class distribution, another important factor affecting performance is the loss function which directs the network to optimise its weights during the training process. The current best performing loss functions can be roughly divided into two types: the loss functions based on Euclidean distance and the loss functions based on Cosine distance. Most of them are derived from Softmax Loss by adding a penalty or modifying softmax directly.

The loss functions based on Euclidean distance include Contrastive Loss [10], Triplet Loss [11], Centre Loss [12], Range Loss [9], and Marginal Loss [13]. These functions are aimed at improving the discriminative ability of features by maximising the inter-class distance or minimising the intra-class distance. Contrastive Loss requires that the network takes two types of sample pairs as inputs – the positive sample pairs (two faces from the same class) and the negative sample pairs (two face images from the different classes). Contrastive Loss minimises the Euclidean distance of the positive pairs and penalises the negative pairs that have a distance smaller than a threshold. Triplet Loss uses the triplet as the input which includes a positive sample, a negative sample and an anchor. An anchor is also a positive sample, which is initially closer to some negative samples than it is to some positive samples. During the training, the anchor-positive pairs are pulled together while the anchor-negative pairs are pushed apart as much as possible. However, the selection of the sample pairs and the triplets is laborious and time-consuming for both Contrastive Loss and Triplet Loss. Centre Loss, Marginal Loss and Range Loss add another penalty to implement the joint supervision with Softmax Loss. Specifically, Centre Loss adds a penalty to Softmax by calculating and restricting the distances between the within-class samples and the corresponding class centre. Marginal Loss considers all the sample pairs in a batch and forces the sample pairs from different classes to have a margin larger than a threshold θ while forcing the samples from the same class to have a margin smaller than the threshold θ. It is however overstrict to force the two farthest samples in a class to have a distance smaller than that of two nearest samples from different classes, which makes the training procedure hard to converge. Range Loss calculates the distances of the samples within each class, and chooses the pair of two samples which have the largest distance as the intra-class constraint; simultaneously, Range Loss calculates the distance of each

pair of class centres (aka centre pair), and forces the centre pair that has the smallest distance to have a larger margin than the designated threshold. However, only considering one centre pair each time is not comprehensive, as more centre pairs may have margins smaller than the designated threshold and thus the training procedure is hard to completely converge because of the slow learning speed.

The loss functions based on Cosine distance include L_2-Softmax Loss [14], L-Softmax Loss [15], A-Softmax Loss [16], AM-Softmax Loss [17], and ArcFace [18]. Based on Softmax loss, L_2-Softmax Loss restricts the L2-norm of the feature descriptor to a constant value. L_2-Softmax Loss brings better geometrical interpretation and pays similar attention to both good and bad quality faces. L-Softmax reformulates the output of softmax layer from $W \cdot f$ to $|W| \cdot |f| \cdot cos\theta$ so as to transform the Euclidean distance to Cosine distance, and also add multiplicative angular constraints to $cos\theta$ to enlarge the angular margins between different identities. Based on L-Softmax Loss, A-Softmax applies weight normalisation, so $W \cdot f$ is further reformulated to $|f| \cdot cos\theta$ which simplifies the training target. However, after using the same multiplicative angular constraints, both L-Softmax and A-Softmax Loss are difficult to converge. So annealing optimization strategy is adopted by these two methods to help the algorithm to converge. To improve the convergence of A-Softmax, Wang et al. [17] propose AM-Softmax which replaces the multiplicative angular constraints with the additive angular constraints, namely, transforms $cos(m\theta)$ to $cos\theta - m$. Besides, AM-Softmax also applies feature normalisation and introduces the global scaling factor $s = 30$ which makes $|W| \cdot |f| = s$. Hence, the training target $|W| \cdot |f| \cdot cos\theta$ is again simplified to $s \cdot cos\theta$. ArcFace also utilises the additive angular constraints, but it changes $cos(m\theta)$ to $cos(\theta + m)$ which makes it have better geometric interpretation. Both AM-Softmax and ArcFace adopt weight normalisation and feature normalisation which restrict all the features to lie on a hypersphere. However, is it overstrict to force all the features to lie on a hypersphere instead of a wider space? Why and how do weight normalisation and feature normalisation benefit the training procedure? These questions are difficult to answer explicitly, and some evidence shows that "soft" feature normalisation may lead to better results [19].

The existing loss functions do not take the margin bias problem into account. To

rectify this margin bias, we propose to set a minimum margin for all pairs of classes, and then design a loss function based on the minimum margin. Inspired by Softmax Loss, Centre Loss and Marginal Loss, we propose a new loss function, *Minimum Margin Loss* (MML), in this paper which aims at forcing all the class centre pairs to have a distance larger than the specified minimum margin. Different from Range Loss, MML penalises all the 'unqualified' class centre pairs instead of only penalising the centre pair that has the smallest distance. MML reuses the centre positions constantly updated by Centre Loss, and directs the training process by joint supervision with Softmax Loss and Centre Loss. To the best of our knowledge, there is no loss function which considers setting a minimum margin between the class centres. However, it is necessary to have such a constraint to rectify the margin bias introduced by class imbalance in training data. To prove the effectiveness of the proposed method, experiments are conducted on seven public datasets – Labeled Faces in the Wild (LFW) [20], Similar-looking LFW (SLLFW) [21], YouTube Faces (YTF) [22], Megaface [7], FaceScrub [23], IJB-B [24] and IJB-C [25]. Results show that MML achieved better performance than Softmax Loss, Centre Loss, Range Loss and Marginal Loss with almost no increase in computing cost. It also achieved competitive performance compared with the state-of-the-art methods.

2. From Softmax Loss to Minimum Margin Loss

2.1. Softmax Loss and Centre Loss

Softmax Loss is the most commonly used loss function, which is presented below:

$$\mathcal{L}_S = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{W_{y_i}^T f_i + b_{y_i}}}{\sum_{j=1}^{K} e^{W_j^T f_i + b_j}} \tag{1}$$

where N is the batch size, K is the class number of a batch, $f_i \in R^d$ denotes the feature of the ith sample belonging to the y_ith class, $W_j \in R^d$ denotes the jth column of the weight matrix W in the final fully connected layer and b_j is the bias term of the jth class. From Eq(1), it can be seen that Softmax Loss is designed to minimise the differences between the predicted labels and the true labels, which in other words means

the target of Softmax Loss is only to separate the features from different classes in the training set instead of learning discriminative features. Such a target is appropriate for close-set tasks, like most application scenarios of object recognition and behaviour recognition. But the application scenarios of face recognition are open-set tasks in most cases, so the discriminative ability of features has considerable influence on the performance of a face recognition system. To enhance the discriminative ability of features, Wen et al. [12] proposed the Centre Loss to minimise the intra-class distance, as shown below:

$$\mathcal{L}_C = \frac{1}{2}\sum_{i=1}^{N} ||f_i - c_{y_i}||_2^2 \tag{2}$$

where c_{y_i} denotes the class centre of the y_ith class. Centre Loss calculates all the distances between the class centres and within-class samples, and is used in conjunction with Softmax Loss:

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_S + \lambda\mathcal{L}_C & (3)\\ &= -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{W_{y_i}^T f_i + b_{y_i}}}{\sum_{j=1}^{K} e^{W_j^T f_i + b_j}} + \frac{\lambda}{2}\sum_{i=1}^{N} ||f_i - c_{y_i}||_2^2 & (4) \end{aligned}$$

where λ is the hyper-parameter for balancing the two loss functions.

2.2. Marginal Loss and Range Loss

After combining Softmax Loss with Centre Loss, the within-class compactness is significantly enhanced. But it is not enough to only use Softmax Loss as the inter-class constraint, as it only encourages the separability of features. So Deng et al. [13] proposed Marginal Loss which also takes the way of joint supervision with the Softmax Loss:

$$\mathcal{L} = \mathcal{L}_S + \lambda\mathcal{L}_{Mar} \tag{5}$$

$$\mathcal{L}_{mar} = \frac{1}{N^2 - N}\sum_{i,j,i\neq j}^{N}\left(\xi - y_{ij}\left(\theta - \left|\left|\frac{f_i}{||f_i||} - \frac{f_j}{||f_j||}\right|\right|_2^2\right)\right)_+ \tag{6}$$

where f_i and f_j are the features of the ith and jth samples in a batch, respectively; $y_{ij} \in \{\pm 1\}$ indicates whether f_i and f_j belong to the same class, $(u)_+$ is defined as

$max(u, 0)$, θ is the threshold to separate the positive pairs and the negative pairs, and ξ is the error margin besides the classification hyperplane.

Marginal Loss considers all the possible combinations of the sample pairs in a batch and specifies a threshold θ to constrain all these sample pairs including the positive pairs and the negative pairs. Marginal Loss forces the distances of the positive pairs to be close up to the threshold θ while forcing the distances of the negative pairs to be farther than the threshold θ. But utilising the same threshold θ to constrain both the positive and negative pairs is not proper. Because it is often the case that the two farthest samples in a class have a distance larger than the two nearest samples of the two different but closest classes. Forcibly changing this situation will make the training procedure hard to converge.

Similar to the aforementioned methods, the Range Loss proposed by Zhang et al. [9] also works with softmax Loss as the supervisory signals:

$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_R \tag{7}$$

Different from Marginal Loss, Range Loss consists of two independent losses, namely $\mathcal{L}_{R_{intra}}$ and $\mathcal{L}_{R_{inter}}$ to calculate the intra-class loss and inter-class loss respectively (see Eq.(8)).

$$\mathcal{L}_R = \alpha \mathcal{L}_{R_{intra}} + \beta \mathcal{L}_{R_{inter}} \tag{8}$$

where α and β are two weights for adjusting the influence of $\mathcal{L}_{R_{intra}}$ and $\mathcal{L}_{R_{inter}}$. Mathematically, $\mathcal{L}_{R_{intra}}$ and $\mathcal{L}_{R_{inter}}$ are defined as follows:

$$\mathcal{L}_{R_{intra}} = \sum_{i \subseteq K} \mathcal{L}^i_{R_{intra}} = \sum_{i \subseteq I} \frac{n}{\sum_{j=1}^{n} \frac{1}{D_{ij}}} \tag{9}$$

$$\mathcal{L}_{R_{inter}} = max(M - D_{Centre}, 0) \tag{10}$$

$$= max(M - ||\overline{x}_{\mathcal{Q}} - \overline{x}_{\mathcal{R}}||_2^2, 0) \tag{11}$$

where K is the class number in current batch, D_{ij} is the jth largest distance of the sample pairs in class i, D_{Centre} is the central distance of two nearest classes in current batch, $\overline{x}_{\mathcal{Q}}$ and $\overline{x}_{\mathcal{R}}$ denote the class centres of class $x_{\mathcal{Q}}$ and $x_{\mathcal{R}}$ which have the shortest central distance, and M is the margin threshold. $\mathcal{L}_{R_{intra}}$ measures all the sample pairs in a class and selects n sample pairs that have the large distances to build the loss for

controlling the within-class compactness. As described in [13], experiments show that $n = 2$ is the best choice. $\mathcal{L}_{R_{inter}}$ aims at forcing the class centre pair that has the smallest distance to have a larger margin up to the designated threshold. But there are more centre pairs that may have distances smaller than the designated threshold. It is not comprehensive enough for only considering one centre pair each time which leads the training procedure to take a long time to completely converge because of the low learning speed.

2.3. The Proposed Minimum Margin Loss

Inspired by Softmax Loss, Centre Loss and Marginal Loss, we propose the Minimum Margin Loss (MML) in this paper. MML is used in conjunction with Softmax Loss and Centre Loss, where Centre Loss is utilised to enhance the within-class compactness, Softmax and MML are applied for improving the between-class separability. Specifically speaking, Softmax is in charge of guaranteeing the correctness of classification while MML aims at optimising the between-class margins. The total loss is shown below:

$$\mathcal{L} = \mathcal{L}_S + \alpha\mathcal{L}_C + \beta\mathcal{L}_M \tag{12}$$

where α and β are the hyper-parameters for adjusting the impact of Centre Loss and MML.

MML specifies a threshold called Minimum Margin. By reusing the class centre positions updated by Centre Loss, MML filters all the class centre pairs based on the specified Minimum Margin. For those pairs which have distances smaller than the threshold, corresponding penalties are added into to the loss value. The detail of MML is formulated as follows:

$$\mathcal{L}_M = \sum_{i,j=1}^{K} max(||c_i - c_j||_2^2 - \mathcal{M}, 0) \tag{13}$$

where K is the class number of a batch, c_i and c_j denote the class centres of the ith and jth classes respectively, and $\mathcal{M}$ represents the designated minimum margin. In each training batch, the class centres are updated by Centre Loss with the following

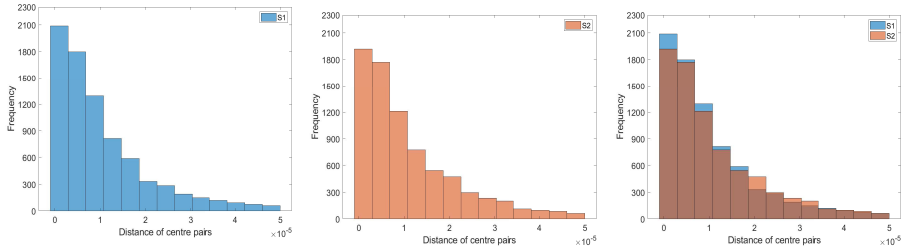
two equations:

$$c_j^{t+1} = c_j^t - \gamma \Delta c_j^t \tag{14}$$

$$\Delta c_j^t = \frac{\sum_{i=1}^{m} \delta(y_i = j)(c_j - f_i)}{1 + \sum_{i=1}^{m} \delta(y_i = j)} \tag{15}$$

where γ is the learning rate of the class centres, t is the number of iteration and $\delta(condition)$ is a conditional function. If the condition is satisfied, $\delta(condition) = 1$, otherwise $\delta(condition) = 0$. Please note that, in Range Loss, the centre of a class is computed by averaging the samples of this class in a batch. However, the size of a batch is limited, and the sample number of a certain class is more limited. Therefore, the class centres generated in this way are not precise compared with the real class centres. Compared with Range Loss, the learned class centres of MML are closer to the real class centres.

Algorithm 1 shows the basic learning steps in the CNNs with the proposed $\mathcal{L}_S + \mathcal{L}_C + \mathcal{L}_M$.



(a) Without using MML (b) After using MML (c) Comparison between S1 and S2

Figure 1: For each class in VGGFace2, its corresponding nearest neighbour class can be found by comparing the positions of different class centres. (a), (b) and (c) show the distributions of the distances between every class centre and its corresponding nearest class centre. Specifically, (a) shows the distribution in the case of using the features generated by Scheme I (without using MML). (b) shows the distribution in the case of using the features generated by Scheme II (using MML). (c) shows the comparison results of (a) and (b), where S1 and S2 represent Scheme I and Scheme II, respectively.

Algorithm 1 Learning algorithm in the CNNs with the proposed $\mathcal{L}_S + \mathcal{L}_C + \mathcal{L}_M$.

Input: Training samples $\{f_i\}$, initialised parameters θ_C in convolution layers, parameters W in the final fully connected layer, and initialised n class centres $\{c_j | j = 1, 2, ..., n\}$. Learning rate μ^t, hyperparameters α and β, learning rate of the class centres γ and the number of iteration $t \leftarrow 1$.

Output: The parameters θ_C.

1: **while** not converge **do**

2: Calculate the total loss by $\mathcal{L}^t = \mathcal{L}_S^t + \alpha\mathcal{L}_C^t + \beta\mathcal{L}_M^t$.

3: Calculate the backpropagation error $\frac{\partial \mathcal{L}^t}{\partial f_i^t}$ for each sample i by $\frac{\partial \mathcal{L}^t}{\partial f_i^t} = \frac{\partial \mathcal{L}_S^t}{\partial f_i^t} + \alpha\frac{\partial \mathcal{L}_C^t}{\partial f_i^t} + \beta\frac{\partial \mathcal{L}_M^t}{\partial f_i^t}$.

4: Update W by $W^{t+1} = W^t - \mu^t \frac{\partial \mathcal{L}^t}{\partial W^t} = W^t - \mu^t \frac{\partial \mathcal{L}_S^t}{\partial W^t}$.

5: Update c_j for each centre j by $c_j^{t+1} = c_j^t - \gamma\Delta c_j^t$.

6: Update θ_C by $\theta_C^{t+1} = \theta_C^t - \mu^t \sum_i^N \frac{\partial \mathcal{L}^t}{\partial f_i^t}\frac{\partial f_i^t}{\partial \theta_C^t}$.

7: $t \leftarrow t + 1$.

8: **end while**

2.4. Discussion

2.4.1. Whether MML can truly enlarge distances of the closest class centre pairs that are smaller than the specified minimum margin

To verify this point, we use the deep models trained by Scheme I (Softmax Loss + Centre Loss) and Scheme II (Softmax Loss + Centre Loss + MML) to extract the features of all the images from a cleaned version of VGGFace2 dataset [6]. The details of the cleaned dataset and the training process of these two models can be found in 3.1. The difference between Scheme I and Scheme II is that Scheme II employs MML as a part of the supervision signal but Scheme I does not. With the extracted features, we calculate the centre position for each class and then calculate the distance between each class centre and its corresponding closest neighbour class centre. The distributions of the distances of these class centres are shown in Figure 1. Figure 1(a) and Figure 1(b) show the distance distributions of Scheme I and Scheme II, respectively. Figure 1(c) makes a comparison between Scheme I and Scheme II, from which we can see that Scheme II has smaller values on the first five bins while owns larger values on the rest of the bins. This indicates that MML enlarges the distance of some neighbour centre pairs, therefore increases the quantity of the centre pairs having large margin.

2.4.2. Whether MML can truly improve the performance of the model on face recognition

To answer this question, we conduct extensive experiments on different benchmark datasets as illustrated in Section 3. The experimental types include face verification, face identification, image-based recognition and video-based recognition. Results show that the proposed method can beat the baseline methods as well as some state-of-the-art methods.

3. Experiments

In this section, we describe the implementation details of the experiments, investigate the influence of the parameters β and $\mathcal{M}$, and evaluate the performance of the proposed method. The evaluations are conducted on MegaFace [7], FaceScrub [23],

LFW [20], SLLFW [21], YTF [22], IJB-B [24] and IJB-C [25] datasets with face identification and face verification tasks. Face identification and face verification are two main tasks of face recognition. Face verification aims at verifying whether two faces are from the person, answering 'Yes' or 'No', which is a binary classification problem. Face identification is to identifying the ID of a face, answering the exact ID, which is a multi-classification problem.

3.1. Experiment Details

Training data. In all experiments, we use VGGFace2 [6] as our training data. To ensure the reliability and the accuracy of the experimental results, we removed all the face images that might be overlapped with the benchmark datasets. As the label noise in the VGGFace2 is very low, no data cleaning has been applied. The final training dataset contains 3.05M face images from 8K identities.

Data preprocessing. MTCNN [26] is applied to all the face images for landmark location, face alignment and face detection. If face detection fails on a training image, we simply discard it; if it fails on a testing image, the provided landmarks are used instead. All the training and testing images are cropped to 160*160 RGB images. To augment the training data, we also perform random horizontal flipping on the training images. To improve the recognition accuracy, we concatenate the features of the original testing image and its horizontally flipped counterpart. Please note that we did not do data cleaning on all the testing sets involved in the experiments including Megaface dataset[1].

Network settings. Based on Inception-ResNet-v1 [27], we implemented and trained five models by Tensorflow [28] according to five supervision schemes: Softmax Loss, Softmax Loss + Centre Loss, Softmax Loss + Marginal Loss, Softmax Loss + Range Loss and Softmax Loss + Centre Loss + MML. For convenience, we use "Softmax

[1]We notice that whether doing cleaning on MegaFace is controversial, as some researchers think it is unfair for the methods previously tested on non-cleaned dataset (e.g. the discussion on GitHub). However, whether doing the cleaning on MegaFace makes much difference in results. According to the results published by MegaFace team, the best methods that using cleaned data can have an accuracy higher than 99% while the best method (BingMMLab-v1) that using non-cleaned data only has an accuracy of 83.758%.

Loss", "Centre Loss", "Marginal Loss", "Range Loss" and "MML" to represent these five schemes, respectively, in the experimental results. We train these five models on one GPU (GTX 1080 Ti), and we set 90 as the batch size, 512 as the embedding size, 5e-4 as the weight decay and 0.4 as the keep probability of the fully connected layer. The total number of iterations is 275K, costing about 30h. The learning rate is initiated as 0.05 and is divided by 10 every 100K iterations. All schemes use the same parameter settings except that Softmax Loss + Centre Loss + MML loads the trained model of Softmax Loss + Centre Loss as the pre-trained model before training starts, as this way makes the former achieve better recognition performance.[2]

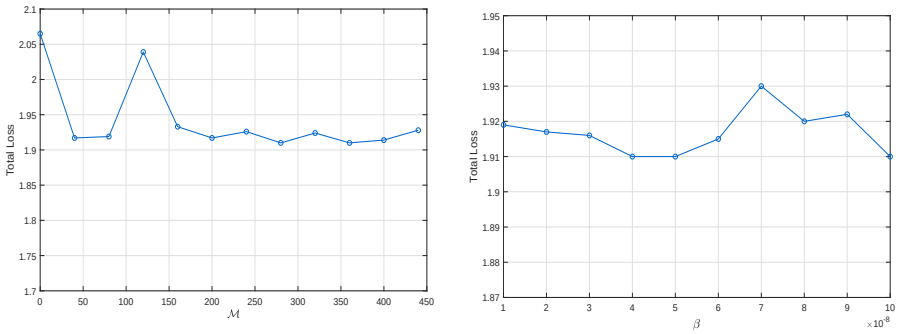
Test settings. During the testing, we try our best to find the parameter settings that lead to highest performance. The α and β in Eq.(12) are set to be 5e-5 and 5e-8, respectively. The minimum margin of MML is set to be 280. The deep feature of each image is obtained from the output of the fully connected layer, and we concatenate the features of the original testing image and its horizontally flipped counterpart, therefore the resulting feature size of each image is $2 * 512$ dimensions. The final verification results are achieved by comparing the threshold with the Euclidean distance of two features

3.2. Influence Analysis on Parameters β and $\mathcal{M}$

β is the hyper-parameter for adjusting the impact of MML in the combination. $\mathcal{M}$ is the designated minimum margin. These two parameters influence the performance of the proposed method. Therefore, how to set these two parameters is a question worthy of study.

[2]Since the training of Softmax Loss + Centre Loss finishes until it fully converges, just reloading the model and resuming training without changing any parameters will not improve the model. In training, the model needs to learn two abilities: the ability to separate different classes (making different classes have no overlap) at the first stage and the ability to enlarge the margin between different classes at the second stage. MML only focuses on the target of the second stage. In addition, MML uses the learned class centres for computing, however the learned class centres cannot reflect the real centres at the early stage as it requires some time for learning. Applying MML at the first stage will cause interference to the training at this stage. Actually, this two-stage training mode can also be regarded as a one-time training by initialising the factor – β to 0 and then setting it to 5e-8 after a certain number of epochs. These two modes are equivalent.

Total loss only reflects the performance of the model on the training set. We conduct two experiments on VGGFace2 dataset and evaluate the influence of these two parameters on total loss. In the first experiment, we fix β to 5e-8, and observe the influence of $\mathcal{M}$ on total loss as shown in Figure 2(a). In the second experiment, we fix $\mathcal{M}$ to 280, and evaluate the relationship between β and total loss as shown in Figure 2(b). From Figure 2(a), we can see that setting $\mathcal{M}$ to 0, namely without using MML, is not proper, as it leads to a high total loss. The lowest total loss appears when $\mathcal{M}$ is 280. From Figure 2(b), we can observe that the total loss remains stable with a wide range of β, but reaches its lowest value when β is 5e-8. Therefore, in the subsequent experiments, we fixed $\mathcal{M}$ and β to 280 and 5e-8, respectively.

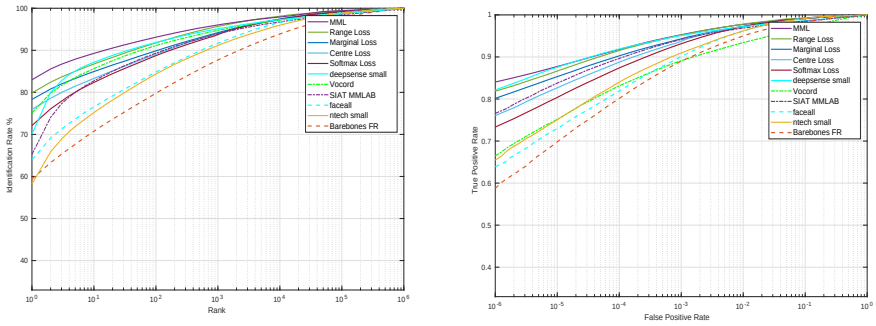


(a) (b)

Figure 2: Face verification accuracies on LFW dataset with two groups of models: (a) fixed β = 5e-8, and different $\mathcal{M}$, (b) fixed $\mathcal{M}$ = 280, and different β.

3.3. MegaFace Challenge 1 on FaceScrub

In this section, we conduct experiment with the MegaFace dataset [7] and the FaceScrub dataset [23]. The MegaFace dataset consists of a million faces and their respective bounding boxes obtained from Flickr (Yahoo's dataset). The FaceScrub dataset is a publicly available dataset containing 0.1M images from 530 identities. According to the experimental protocol of MegaFace Challenge 1, the MegaFace dataset is used as the distractor set, while the FaceScrub dataset is used as the test set. The evaluation is



(a) (b)

Figure 3: (a) reports the CMC curves of different methods with 1M distractors on MegaFace Set 1. (b) reports the ROC curves of different methods with 1M distractors on MegaFace Set 1.

Table 2: The identification rates and the verification rates of different methods on Megaface and FaceScrub datasets with 1M distractors.

Methods	Rank1@10^6	Rank100@10^6	VR @FAR10^{-6}	VR @FAR 10^{-5}
Barebones FR	59.36%	79.79%	58.77%	69.80%
ntech small	58.21%	84.34%	65.48%	75.07%
faceall	63.97%	84.84%	63.89%	72.99%
SIAT MMLAB	65.23%	89.33%	76.56%	83.78%
Vocord	75.13%	91.11%	66.50%	75.15%
deepsense small	70.06%	91.85%	82.15%	87.56%
Softmax Loss	72.11%	88.73%	73.33%	80.37%
Centre Loss	75.93%	89.07%	76.07%	82.66%
Marginal Loss	78.32%	89.87%	80.16%	85.32%
Range Loss	79.86%	91.76%	81.85%	86.65%
MML	**83.00%**	**93.12%**	**84.03%**	**87.73%**

Figure 4: Some examples from the LFW dataset (left) and the YTF dataset (right).

conducted with the officially provided code [7]. More details about the experimental protocol can be found in [7].

We compare the proposed method (MML) with different losses and some deep learning-based methods provided by MegaFace team[3]. In the face identification experiments, the Cumulative Match Characteristics (CMC) curves [29] are calculated to measure the ranking capabilities of different methods, as illustrated by Figure 3(a)). In the face verification experiments, we use the Receiver Operating Characteristic (ROC) curves to evaluate the different methods. The ROC curves plot the False Accept Rate (FAR) of a 1:1 matcher versus the False Reject Rate (FRR) of the matcher which are shown in Figure 3(b). Table 2 lists the numeric results of different methods on identification rates and the verification rates with 1M distractors.

From Figure 3(a), Figure 3(b) and Table 2, we can observe that MML performs better compared with other deep learning-based methods on both identification and verification test. This demonstrates the effectiveness of the whole framework. The proposed MML consistently outperforms Softmax, Centre Loss, Marginal Loss and Range Loss, which confirms the effectiveness of the proposed loss function.

[3]The features of the benchmark methods provided by MegaFace team: http://megaface.cs.washington.edu/participate/challenge.html

Table 3: Verification Rates of state-of-the-art methods on LFW and YTF datasets.

Methods	Images	VR on LFW(%)	VR on YTF(%)
ICCV17' Range Loss [9]	1.5M	99.52	93.7
CVPR17' Marginal Loss [13]	4M	99.48	96.0
BMVC15' VGG Face [30]	2.6M	98.95	97.3
CVPR14' Deep Face [31]	4M	97.35	91.4
ICCV15' FaceNet [11]	200M	99.63	95.1
ECCV16' Centre Loss [12]	0.7M	99.28	94.9
NIPS16' Multibatch [32]	2.6M	98.20	
ECCV16' Aug [33]	0.5M	98.06	
CVPR17' SphereFace [16]	0.5M	99.42	95.0
ECCV18' Contrastive CNN [34]	0.5M	99.12	
ECCV18' OE-CNNs [35]	1.7M	99.47	
Softmax Loss	3.05M	99.43	94.9
Centre Loss	3.05M	99.50	95.1
Range Loss	3.05M	99.50	95.1
Marginal Loss	3.05M	99.52	95.3
MML (Proposed)	3.05M	99.63	95.5

3.4. Comparison with the State-of-the-art Methods on LFW and YTF Datasets

In this section, we evaluate the proposed method on two public benchmark datasets – LFW [20] and YTF [22] datasets according to the settings in Section 3.1. Some preprocessed examples from these two datasets are shown in Figure 4.

LFW dataset is collected from the web, which contains 13,233 face images with large variations in facial paraphernalia, pose and expression. These face images come from 5749 different identities where 4069 of them have one image and the remaining 1680 identities have at least two images. LFW utilises the Viola-Jones face detector, which is the only constraint on the faces collected. We follow the standard experimental protocol of unrestricted with labelled outside data [36] and test 6,000 face pairs according to the given pair list.

YTF dataset consists of 3,425 videos obtained from YouTube. These videos come from 1,595 identities with an average of 2.15 videos for each person. The frame number of the video clips ranges from 48 to 6,070, and the average is 181.3 frames. Also, we follow the standard experimental protocol of unrestricted with labelled outside data to evaluate the performance of the relevant methods on the given 5,000 video pairs.

Table 3 shows the results of the proposed method and the state-of-the-art methods on LFW and YTF datasets, from which we can observe the followings.

- The proposed MML outperforms Softmax Loss and Centre Loss, increasing the verification performance both on LFW and YTF datasets. On LFW, the accuracy improves from 99.43% and 99.50% to 99.63%, while on YTF, the accuracy increases from 94.9% and 95.1% to 95.5%. Also, MML outperforms Range Loss and Marginal Loss both on LFW and YTF datasets. On LFW, the accuracy improves from 99.50% and 99.52% to 99.63%, while on YTF, the accuracy increases from 95.1% and 95.3% to 95.5%. This demonstrates the effectiveness of the MML, also demonstrates the effectiveness of the combination of Softmax Loss + Centre Loss + MML.

- Compared with the state-of-the-art methods, the proposed method has an accuracy of 99.63% on LFW and 95.5% on YTF, higher than most of the methods. FaceNet is neck and neck with the proposed method on LFW, but FaceNet

uses a large scale dataset which includes approximately 200 million face images. Consequently, FaceNet requires much more time for training compared with the proposed method which only uses 3.05 million face images.

3.5. Further Comparison on SLLFW Dataset

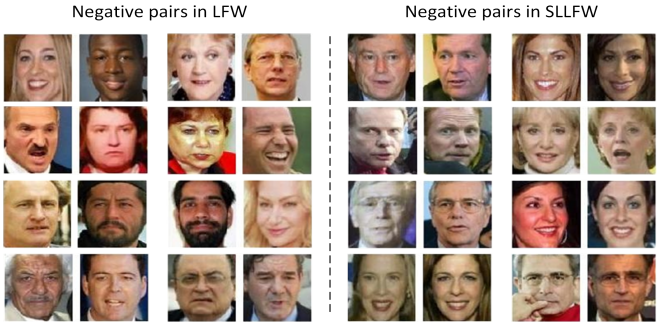


Figure 5: Examples of the negative pairs in LFW and SLLFW. Compared to the negative pairs in LFW, the negative pairs in SLLFW are quite difficult to distinguish.

As more and more methods are gradually touching the theoretical upper limit[4] of LFW, the gaps between different methods become more and more narrow, making it hard to differentiate different methods. Therefore, to confirm the performance of MML, an additional experiment is conducted on SLLFW [21]. SLLFW uses the same positive pairs as LFW for testing, but in SLLFW, 3000 similar-looking face pairs are deliberately selected out from LFW by human crowdsourcing to replace the random negative pairs in LFW. Some examples of the negative pairs in LFW and SLLFW are shown in Fig. 5. Compared with LFW, SLLFW adds more challenges to the testing, causing the accuracy of the same state-of-the-art methods to drop by 10-20%.

Table 4 shows the verification accuracy of different methods on SLLFW. The results of some benchmark methods are shown in the top half of the table. These results are publicly accessible [38] and provided by the SLLFW team[21]. As can be seen from Table 4, MML achieves considerably better performance than the benchmark methods

[4]There are 6 mismatched pairs on LFW which are incorrectly labelled as matched. So the upper limit accuracy on LFW is (6000-6)/6000=99.90%.

Table 4: Verification performance of different methods on SLLFW.

Method	Images	LFW(%)	SLLFW(%)
Deep Face [31]	0.5M	92.87	78.78
DeepID2 [10]	0.2M	95.00	78.25
VGG Face [30]	2.6M	96.70	85.78
DCMN [21]	0.5M	98.03	91.00
Noisy Softmax [37]	0.5M	99.18	94.50
Softmax Loss	3.05M	99.43	95.92
Centre Loss	3.05M	99.50	96.02
Range Loss	3.05M	99.50	96.07
Marginal Loss	3.05M	99.52	96.07
MML	3.05M	99.63	96.37

on SLLFW. Also MML shows higher accuracy than other relevant loss functions. In the top half of the table, the accuracy of the benchmark methods drops only by between 16.75% and 4.68% from LFW to SLLFW. By comparison, the accuracy of MML drops by 3.26%. The results on SLLFW further confirm the performance of the proposed methods.

3.6. Results on IJB-B and IJB-C

The IJB-B dataset [24] is composed of 21.8K still images and 55K frames from 7,011 videos. In IJB-B, there are 1,845 subjects which have no overlap with the popular face recognition benchmarks, such as VGGFace2 [6] and CASIA WebFace [8]. In IJB-B, there are totally 12,115 templates with 10,270 genuine matches and 8M impostor matches. The IJB-C dataset [25] is an extension of IJB-B. It contains 31.3K still images and 117.5K frames from 11,779 videos. All these images and videos are from 3,531 subjects which also have no overlap with the popular face recognition benchmarks. In IJB-C, there are totally 23,124 templates including 19, 557 genuine matches and 15, 639K impostor matches.

Following the 1:1 verification protocol, we compare the proposed MML with the most recent methods as shown in the upper part of Table 5. For a fairer comparison,

Table 5: Evaluation results with 1:1 verification protocol on IJB-B and IJB-C datasets.

Method	IJB-B TAR@FAR=1e-4	IJB-C TAR@FAR=1e-4
Crystal Loss [39]	0.898	0.919
ResNet50 [6]	0.784	0.825
SENet50 [6]	0.800	0.840
ResNet50+SENet50 [6]	0.800	0.841
MN-v [40]	0.818	0.852
MN-vc [40]	0.831	0.862
ResNet50+DCN(Kpts) [41]	0.850	0.867
ResNet50+DCN(Divs) [41]	0.841	0.880
SENet50+DCN(Kpts) [41]	0.846	0.874
SENet50+DCN(Divs) [41]	0.849	0.885
GAN+ArcFace [42]	0.904	0.926
PCP+ArcFace [42]	0.901	0.924
PCPSM+ArcFace [42]	0.907	0.928
LRR+ArcFace [42]	0.909	0.931
PCPSFM+ArcFace [42]	0.911	0.934
Softmax Loss	0.908	0.931
Centre Loss	0.910	0.934
Range Loss	0.916	0.937
Marginal Loss	0.917	0.939
MML	**0.921**	**0.943**

we also directly compare MML with other popular and relevant loss functions under the same framework. Results show that MML performs better than the most recent methods as shown in the upper part of Table 5 on both IJB-B and IJB-C datasets. Also, MML shows better performance than the relevant loss functions compared in the lower part of Table 5.

4. Conclusion

In this paper, a new loss function – Minimum Margin Loss (MML) is presented to guide deep neural networks to learn highly discriminative face features. To the best of our knowledge, MML is the first loss that considers setting a minimum margin between the different classes. We show that the proposed loss function is very easy to implement in the CNNs and our CNN models can be directly optimized by the standard SGD. Extensive experiments are conducted on the seven public available datasets. We compare MML with the methods published in the past few years on top conference and journals. We also directly compare MML with the relevant loss functions under the same framework. Results show that MML has state-of-the-art performance. Future research is needed to automatically determine the minimum margin $\mathcal{M}$. Also we will try to give the theoretical proof about the advantage of setting a minimum margin in the future work.

References

[1] J. M. Pandya, D. Rathod, J. J. Jadav, A survey of face recognition approach, International Journal of Engineering Research and Applications (IJERA) 3 (1) (2013) 632–635.

[2] L. Wu, Y. Wang, J. Gao, X. Li, Deep adaptive feature embedding with local sample distributions for person re-identification, Pattern Recognition 73 (2018) 275–288.

[3] J. Han, D. Zhang, G. Cheng, N. Liu, D. Xu, Advanced deep-learning techniques for salient and category-specific object detection: a survey, IEEE Signal Processing Magazine 35 (1) (2018) 84–100.

[4] M. Ma, N. Marturi, Y. Li, A. Leonardis, R. Stolkin, Region-sequence based six-stream CNN features for general and fine-grained human action recognition in videos, Pattern Recognition 76 (2018) 506–521.

[5] Y. Guo, L. Zhang, Y. Hu, X. He, J. Gao, MS-Celeb-1M: A dataset and benchmark for large scale face recognition, in: European Conference on Computer Vision, Springer, 2016.

[6] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, A. Zisserman, Vggface2: A dataset for recognising faces across pose and age, in: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), IEEE, 2018, pp. 67–74.

[7] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, E. Brossard, The megaface benchmark: 1 million faces for recognition at scale, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4873–4882.

[8] D. Yi, Z. Lei, S. Liao, S. Z. Li, Learning face representation from scratch, arXiv preprint:1411.7923v1, 28 Nov 2014.

[9] X. Zhang, Z. Fang, Y. Wen, Z. Li, Y. Qiao, Range Loss for Deep Face Recognition with Long-Tailed Training Data, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 5419–5428.

[10] Y. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: Advances in neural information processing systems, 2014, pp. 1988–1996.

[11] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.

[12] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A Discriminative Feature Learning Approach for Deep Face Recognition, in: Computer Vision – ECCV 2016, Lecture Notes in Computer Science, Springer, Cham, 2016, pp. 499–515.

[13] J. Deng, Y. Zhou, S. Zafeiriou, Marginal loss for deep face recognition, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 2006–2014.

[14] R. Ranjan, C. D. Castillo, R. Chellappa, L2-constrained softmax loss for discriminative face verification, arXiv preprint:1703.09507v3, 7 Jun 2017.

[15] W. Liu, Y. Wen, Z. Yu, M. Yang, Large-Margin Softmax Loss for Convolutional Neural Networks, in: International Conference on Machine Learning, 2016, pp. 507–516.

[16] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, L. Song, Sphereface: Deep hypersphere embedding for face recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 212–220.

[17] F. Wang, J. Cheng, W. Liu, H. Liu, Additive margin softmax for face verification, IEEE Signal Processing Letters 25 (7) (2018) 926–930.

[18] J. Deng, J. Guo, S. Zafeiriou, Arcface: Additive angular margin loss for deep face recognition, arXiv preprint:1801.07698v3, 9 Feb 2019.

[19] Y. Zheng, D. K. Pal, M. Savvides, Ring loss: Convex feature normalization for face recognition, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[20] G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: A database for studying face recognition in unconstrained environments, Tech. Rep. 07-49, University of Massachusetts, Amherst (October 2007).

[21] W. Deng, J. Hu, N. Zhang, B. Chen, J. Guo, Fine-grained face verification: Fglfw database, baselines, and human-dcmn partnership, Pattern Recognition 66 (2017) 63–73.

[22] L. Wolf, T. Hassner, I. Maoz, Face recognition in unconstrained videos with matched background similarity, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 529–534.

[23] H.-W. Ng, S. Winkler, A data-driven approach to cleaning large face datasets, in: Image Processing (ICIP), 2014 IEEE International Conference on, IEEE, 2014, pp. 343–347.

[24] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen, J. Cheney, P. Grother, Iarpa janus benchmark-b face dataset, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2017.

[25] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, P. Grother, Iarpa janus benchmark - c: Face dataset and protocol, in: 2018 International Conference on Biometrics (ICB), 2018, pp. 158–165. doi:10.1109/ICB2018.2018.00033.

[26] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint face detection and alignment using multitask cascaded convolutional networks, IEEE Signal Processing Letters 23 (10) (2016) 1499–1503.

[27] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[28] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint:1603.04467v2, Mar 2016.

[29] P. J. Phillips, P. Grother, R. J. Micheals, D. M. Blackburn, E. Tabassi, M. Bone, R. V. T. FACE, Evaluation report, Facial Recognit. Vendor Test 2002.

[30] O. M. Parkhi, A. Vedaldi, A. Zisserman, Deep Face Recognition., in: 2015 British Machine Vision Conference (BMVC), Vol. 1, 2015, p. 6.

[31] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, DeepFace: Closing the Gap to Human-Level Performance in Face Verification, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '14, IEEE Computer Society, Washington, DC, USA, 2014, pp. 1701–1708.

[32] O. Tadmor, T. Rosenwein, S. Shalev-Shwartz, Y. Wexler, A. Shashua, Learning a Metric Embedding for Face Recognition Using the Multibatch Method, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, Curran Associates Inc., USA, 2016, pp. 1396–1397.

[33] I. Masi, A. T. Tran, T. Hassner, J. T. Leksut, G. Medioni, Do We Really Need to Collect Millions of Faces for Effective Face Recognition?, in: Computer Vision – ECCV 2016, Lecture Notes in Computer Science, Springer, Cham, 2016, pp. 579–596.

[34] C. Han, S. Shan, M. Kan, S. Wu, X. Chen, Face recognition with contrastive convolution, in: The European Conference on Computer Vision (ECCV), 2018.

[35] Y. Wang, D. Gong, Z. Zhou, X. Ji, H. Wang, Z. Li, W. Liu, T. Zhang, Orthogonal deep features decomposition for age-invariant face recognition, in: The European Conference on Computer Vision (ECCV), 2018.

[36] G. B. Huang, E. Learned-Miller, Labeled faces in the wild: Updates and new reporting procedures, Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep (2014) 14–003.

[37] B. Chen, W. Deng, J. Du, Noisy softmax: Improving the generalization ability of dcnn via postponing the early softmax saturation, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[38] The results of some baseline methods provided by SLLFW team:.
URL `http://www.whdeng.cn/SLLFW/#results`

[39] R. Ranjan, A. Bansal, J. Zheng, H. Xu, J. Gleason, B. Lu, A. Nanduri, J.-C. Chen, C. D. Castillo, R. Chellappa, A fast and accurate system for face detection, identification, and verification, IEEE Transactions on Biometrics, Behavior, and Identity Science 1 (2) (2019) 82–96.

[40] W. Xie, A. Zisserman, Multicolumn networks for face recognition, arXiv preprint:1807.09192v1, 24 Jul 2018.

[41] W. Xie, L. Shen, A. Zisserman, Comparator networks, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 782–797.

[42] N. Xue, J. Deng, S. Cheng, Y. Panagakis, S. P. Zafeiriou, Side information for face completion: a robust pca approach, IEEE Transactions on Pattern Analysis and Machine Intelligence (2019) 1–1doi:10.1109/TPAMI.2019.2902556.